In the *24th International Symposium on Robot and Human Interactive Communication* (RO-MAN), Kobe, Japan, August 31-September 4, 2015.

Incremental Knowledge Acquisition for Human-Robot Collaboration

Batbold Myagmarjav
Department of Computer Science
Texas Tech University
Lubbock TX 79409, USA
Email: bat.myagmarjav@ttu.edu

Mohan Sridharan
Department of Electrical and Computer Engineering
The University of Auckland
Auckland 1142, New Zealand
Email: m.sridharan@auckland.ac.nz

Abstract— **Human-robot collaboration in practical domains typically requires considerable domain knowledge and labeled examples of objects and events of interest. Robots frequently face unforeseen situations in such domains, and it may be difficult to provide labeled samples. Active learning algorithms have been developed to allow robots to ask questions and acquire relevant information when necessary. However, human participants may lack the time and expertise to provide comprehensive feedback. The incremental active learning architecture described in this paper addresses these challenges by posing questions with the objective of maximizing the potential utility of the response from humans who lack domain expertise. Candidate questions are generated using contextual cues, and ranked using a measure of utility that is based on measures of information gain, ambiguity and human confusion. The top-ranked questions are used to update the robot's knowledge by soliciting answers from human participants. The architecture's capabilities are evaluated in a simulated domain, demonstrating a significant reduction in the number of questions posed in comparison with algorithms that use the individual measures or select questions randomly from the set of candidate questions.**

Index Terms— **Human-robot interaction, incremental knowledge acquisition, contextual query generation.**

I. Introduction

Robots[1] collaborating with humans in complex domains typically need a significant amount of domain knowledge. It is, however, difficult to equip robots with accurate and complete domain knowledge, and human participants may lack the expertise and time to provide elaborate instructions to robots. The ability to pose relevant questions that quickly draw a human's attention to the object(s) and event(s) of interest can thus significantly influence the quality of a robot's interaction with humans.

Humans frequently use contextual cues to draw attention to an object of interest. Such contextual information is all the more useful when we forget the word(s) typically used to describe an object, or if our collaborator does not have the necessary background knowledge to understand our description. Contextual cues can take different forms, and positional context with reference to a known object can be very useful in disambiguating the object of interest. For instance, instead of referring to a "1965 Ford Mustang" in a busy street intersection, we may refer to the "red car behind the bus", using both feature labels (e.g., color and object labels) and positional reference to a known object. Humans also incrementally learn from, and build upon, existing knowledge, by posing questions to acquire information from parents, teachers and friends. Furthermore, since we may often be embarrassed to ask "stupid questions", we attempt to formulate interesting questions that help us quickly acquire the desired information. Consider, for instance, the common question: "what is that?", which even in the presence of other cues (e.g., gestures) is likely to provide an ambiguous reference to the person we are interacting with, resulting in a possibly inaccurate response. In contrast, the question: "what is in your right hand?" is more likely to obtain an accurate answer by unambiguously drawing attention to the object of interest. Motivated by these instinctual choices made by humans, this paper describes an architecture for incremental knowledge acquisition from visual and verbal cues in human-robot interaction. An agent equipped with this architecture:

- Constructs candidate questions about objects in a scene under consideration, based on current domain knowledge and the contextual information available for use.
- Ranks these questions based on their relative utility, i.e., their ability to minimize interaction with humans; utility is computed using heuristic measures of information gain, ambiguity and human confusion.
- Solicits human feedback by posing the top-ranked questions, updating knowledge and incrementally learning about the objects and scenes under consideration.

We illustrate and evaluate the capabilities of the architecture on simulated images of scenes with objects characterized by different colors and shapes. The robot's objective is to start with incomplete knowledge about objects in the scene, and learn the labels of the objects and features in the scene by posing as few questions as possible. Although this objective and the simulated domain may appear simplistic, they (a) capture the research challenges of interest, which are also intrinsic to more complex human-robot collaboration domains; and (b) help isolate and thoroughly analyze the contributions of the proposed algorithms and measures.

The remainder of the paper is organized as follows. Section II reviews a representative set of related work. Section III describes the proposed architecture and its components. Section IV describes the experimental setup and discusses the results of experimental evaluation. Finally, Section V presents the conclusions along with future plans.

[1] In this paper, "agent", "robot" and "learner" are used interchangeably.

II. Related Work

This section motivates the proposed architecture by reviewing a representative set of related work.

Active learning algorithms allow incremental labeling or acquisition of data, e.g., by allowing a human *annotator* to label instances in the dataset that have been misclassified by existing models. A recent survey categorized active learning algorithms into *pool-based*, *stream-based* and *membership query* algorithms [1]. Existing algorithms predominantly focus on choosing unlabeled instances that are to be presented to the annotator, rather than evaluating the types of queries to ask [1], [2], [3]. However, research indicates that the introduction of feature queries allowing labeling of features as well as object instances significantly improves performance based on the learned models [4].

Active learning has been combined with multiple instance learning (MIL) to minimize human supervision by supporting the labeling of bags (e.g., images) instead of individual instances (e.g., objects and features in the images) [5]. In such combinations, research shows that an incremental learning architecture that provides the ability to solicit labels of previously unseen bags, speeds up learning of object models and results in more accurate object recognition based on these models [6]. Research also shows that a multimodal learning algorithm that associates visual features with verbal descriptions (provided by humans) leads to object models that provide more accurate object recognition than models based on just visual features [7]. Although these algorithms reduce human involvement, the focus is on labeling bags and not on the relative merits of different query types.

AI researchers have developed algorithms that allow the learners to ask different types of questions. Research has explored the embedding of context in queries to improve the overall quality of the questions posed by a robot to human participants [8]. However, this algorithm evaluated the reaction of humans, and the ability of humans to answer questions correctly, instead of the agent's ability to learn from these questions. A different approach to asking the right questions were explored in [9], where a decision tree was used to identify a series of questions that would extract the desired information. Another option is to pose query generation as a planning task, but it requires prior knowledge of possible answers, which will differ from scene to scene; the planning will also be computationally inefficient.

Learning from demonstration (LfD) algorithms allow agents to observe a human demonstrate a specific task, and either mimic the observed actions or map the actions to available capabilities. Common algorithms that use teleoperation, planning and demonstration learning techniques have been surveyed and discussed in [10], [11]. More recent research has combined active learning with LfD to explore the use of different types of questions [12]. This work introduced four types of queries: *object label*, *feature label*, *demonstration*, and *affirmation* queries, but the objective was to explore how each query category is perceived by humans.

Our proposed architecture seeks to address the limitations of existing work by allowing the learner to use contextual cues and incrementally pose questions with high relative utility, i.e., questions that help disambiguate between (and quickly acquire information about) domain objects.

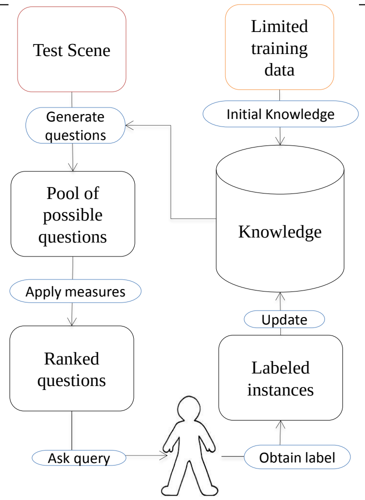

Fig. 1. Overview of the proposed architecture.

III. Problem Formulation

Figure 1 is an overview of the proposed architecture. The architecture starts with limited knowledge of the *scene*. The set of possible queries are generated using contextual cues, as described in Section III-A. Candidate queries are ranked based on measures of information gain, ambiguity, and human confusion (Section III-B), and the top-ranked queries are posed to a human annotator. Human input is used to revise domain knowledge, which is used to generate a new set of queries until all objects and features are labeled. The following notation will be used throughout this paper:

1. An object in the domain can be characterized by n different properties or *features*.
2. $\mathcal{F} = \{\mathcal{F}_1, ..., \mathcal{F}_n\}$ denotes a superset of features. Each $\mathcal{F}_i, i \in [1.n]$, is a set of instances of one type of feature (e.g., color).
3. A feature instance $f \in \mathcal{F}_i$ is a tuple $\langle label, values \rangle$, where $label$ refers to a human understandable word, e.g., *red*, and $values$ refer to the computer representation of that label, e.g., RGB value $(255, 0, 0)$.
4. *Scene* $\mathcal{S}$ is a set of objects. Each $s \in \mathcal{S}$ is the tuple $\langle label, \mathcal{OF} \rangle$, where $\mathcal{OF} = \{f_1, ..., f_n\}$ with $f_i \in \mathcal{F}_i, i \in [1, n]$. Each object thus has a label and one instance of each feature, e.g., *black* for color and *circle* for shape. Object s with feature f_i is denoted by $f_i(s)$.
5. $\mathcal{R}$ is the set of *Relations* that can exist between two objects in a scene. Each relation $r \in \mathcal{R}$ between two objects in the *scene* is assumed to be determinable and their labels are known to the architecture e.g., relative positions of two objects in space, and the temporal relations between two events. Such relations are denoted by $r(s_i, s_j)$, where $s_i, s_j \in \mathcal{S}$ and $s_i \neq s_j$.

6. *Knowledge Base* denoted as $\mathcal{K}$ is the tuple defined as $\langle \mathcal{S}, \mathcal{LS}, \mathcal{US}, \mathcal{LF}, \mathcal{UF}, \mathcal{R} \rangle$:
   - $\mathcal{LS}$ is the set of labeled scene objects, with $\mathcal{LS} \subseteq \mathcal{S}$.
   - $\mathcal{US}$ denotes the set of unlabeled scene objects such that $\mathcal{US} \subseteq \mathcal{S}$. Note that $\mathcal{LS} \cap \mathcal{US} \equiv \varnothing$, i.e., these sets have no common members.
   - $\mathcal{LF}$ denotes the superset of labeled features: $\{\mathcal{LF}_1, \mathcal{LF}_2, ..., \mathcal{LF}_n\}$. Each $\mathcal{LF}_i \subseteq \mathcal{F}_i, i \in [1, n]$ contains instances of a specific feature (e.g., color) with known labels.
   - $\mathcal{UF}$ denotes the superset of unlabeled features: $\{\mathcal{UF}_1, \mathcal{UF}_2, ..., \mathcal{UF}_n\}$. Each $\mathcal{UF}_i \subseteq \mathcal{F}_i, i \in [1, n]$ contains the instances of a specific feature with unknown labels. Note that $\mathcal{LF}_i \cap \mathcal{UF}_i \equiv \varnothing$.

Using this notation, we next describe the steps for generating candidate queries (Section III-A), and the measures for ranking the queries (Section III-B).

A. Query Generation

This section describes the generation of a set of candidate queries $\mathcal{Q}$ for a scene, where each query $q \in \mathcal{Q}$ contains embedded contextual information to describe the object of interest. Specifically $q = \langle t, s, \mathcal{C} \rangle$, where:

- t denotes the query type e.g., object label query or feature label query.
- $s \in \mathcal{S}$ denotes the object of interest in the scene.
- $\mathcal{C}$ denotes the embedded context, which describes the object of interest. Specifically $\mathcal{C} = \langle \mathcal{SC}, \mathcal{LC}, gc \rangle$ where:
  - $\mathcal{SC}$ denotes the set of *self contexts*. The labeled feature(s) of s or the label of s can be a self context.
  - $\mathcal{LC}$ denotes the set of *local contexts*. Local contexts are labeled objects or features that are related to s i.e., $r(s, s_i)$ such that $s_i \in \mathcal{S}$, $r \in \mathcal{R}$, $s_i \neq s$, with $s_i \in \mathcal{LS}$ or $\exists f \in \mathcal{LF}$ such that $f(s_i)$.
  - gc denotes the *global context* defined by its relation to the whole scene. We assume that only one global context exists for each object (e.g., *top right corner*), and that gc is computable.

Algorithm 1 describes the generation of queries with *level-1* context. The input is a scene object s and the output is a set of possible questions $\mathcal{Q}$. First, all the context which can describe s is retrieved. Note that global context gc is computed using predefined subroutines. Self contexts $\mathcal{SC}$ of the object s are the known object label or labeled features (e.g., *red*) of s. Local contexts are the labeled objects or features that are related to s (e.g., *above* red object). If the object label of s is unknown, i.e., $s \in \mathcal{US}$, an object label query is generated using the global context and added to $\mathcal{Q}$. Object label queries are also generated using each of the self contexts of s and added to $\mathcal{Q}$. Next, object label queries are generated for s using each of its local contexts, and added to $\mathcal{Q}$. It is possible that no context exists to describe s, resulting in zero queries about unlabeled components of s. After object queries are generated and collected, each feature of s is checked for labels. Using the same global context used above, each unlabeled feature generates a feature label query to be added to $\mathcal{Q}$. Similarly, feature label queries are generated using the self contexts of s, and the local contexts of s, and added to $\mathcal{Q}$. Finally, $\mathcal{Q}$ is returned as output.

```
Algorithm 1: Level 1 Query Generation
Input: $s$: a scene object, and knowledge base
Output: $\mathcal{Q}$: set of queries
Procedure QueryGeneration()
$\mathcal{C} \leftarrow Context(s)$
$Initialize\ \mathcal{SC}\ with\ \mathcal{C}[0]$
$Initialize\ \mathcal{LC}\ with\ \mathcal{C}[1]$
$Initialize\ gc\ with\ \mathcal{C}[2]$
if $s \in \mathcal{US}$ then
    $q \leftarrow \langle object, s, \langle \varnothing, \varnothing, gc \rangle \rangle$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
    for each $sc \in \mathcal{SC}$ do
        $q \leftarrow \langle object, s, \langle \{sc\}, \varnothing, null \rangle \rangle$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
    end
    for each $lc \in \mathcal{LC}$ do
        $q \leftarrow \langle object, s, \langle \varnothing, \{lc\}, null \rangle \rangle$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
    end
end
for each feature $f$ in $f(s)$ do
    if $f \in \mathcal{UF}$ then
        $q \leftarrow \langle f, s, \langle \varnothing, \varnothing, gc \rangle \rangle$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
        for each $sc \in \mathcal{SC}$ do
            $q \leftarrow \langle f, s, \langle \{sc\}, \varnothing, null \rangle \rangle$
            $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
        end
        for each $lc \in \mathcal{LC}$ do
            $q \leftarrow \langle f, s, \langle \varnothing, \{lc\}, null \rangle \rangle$
            $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
        end
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{q\}$
    end
end
return $\mathcal{Q}$

Procedure Context(s)
$\mathcal{SC} = \{f \in \mathcal{LF} \mid f(s)\}$
if $s \in \mathcal{LS}$ then
    $\mathcal{SC} = \mathcal{SC} \cup \{label\ of\ s\}$
end
$\mathcal{LC} = \{s_i \in \mathcal{S} \mid s_i \neq s, \exists r(s, s_i), s_i \in \mathcal{LS}\}$
$\mathcal{LC} = \mathcal{LC} \cup \{s_i \in \mathcal{S} \mid s_i \neq s, \exists r(s, s_i), f(s_i) : f \in \mathcal{LF}\}$
Compute $gc(s)$                                ▷ predefined subroutine
return $\langle \mathcal{SC}, \mathcal{LC}, gc \rangle$
```

A simplistic question template was used for constructing queries: *<Question word(s)> <Type> <Context>?*
With the following specific example: *<What is the> <color label of the object> <below the cross>?* Self context

information, e.g., *red object*, is an exception to this template; there is no additional context in the corresponding queries.

1) Level of Context: Humans, especially those without domain expertise, are likely to be overwhelmed by a large amount of contextual information, leading to information overload and inaccurate responses. We consider different levels of contextual information, and limit ourselves to three levels, using α to denote *human confusion*. **Level 1** queries are least likely to confuse the human annotator, while **Level 3** queries are the most confusing due to the amount of contextual information considered. The levels are defined as:

- **Level 1**: One item of contextual information.
  - A self context, e.g., *red object*.
  - Two self contexts, e.g., *red rectangle*.
  - A local context, e.g., object *above the red object*.
  - A global context, e.g., *top right corner*.
- **Level 2**: Two items of contextual information.
  - A self context and one global context, e.g., *red object in the top right corner*.
  - A self context and a local context, e.g., *red object above the triangle*.
  - A local context and a global context, e.g., *object at the top, above the red object*.
  - Two local contexts, e.g., *object above the red object, to the right of the circle*.
- **Level 3**: Three items of contextual information.
  - A self context, a global context, and a local context, e.g., *red object in the top right corner, next to the circle*.
  - Two self contexts and a local context. e.g., *red circle above the triangle*.
  - Two self contexts and a global context. e.g., *red circle at the top of the scene*.
  - Two local contexts and a self context. e.g., *red object to the right of the circle, above the green object*.
  - Two local contexts and a global context. e.g., *object on the right, above the circle and to the right of the yellow object*.

Queries of a specific level are generated if the corresponding contextual cue exists, e.g., there are labeled feature or object instances that can be used to describe the object of interest.

B. Query selection

After the set of queries $\mathcal{Q}$ is generated, the most useful queries can be identified as those that (1) maximize information gain; (2) minimize ambiguity; and (3) minimize human confusion. We designed heuristic measures to capture these intuitive principles. The first measure captures the potential information gain if human annotation is obtained for a query $q \in \mathcal{Q}$ (Section III-B.1). The second measure captures how the embedded contextual information in q uniquely describes the object of interest (Section III-B.2). The information obtained from these two measures is combined, using a measure of human confusion to break ties (Section III-B.3).

1) Information Gain: Consider a set of m distinct objects in a *scene*; each object has one instance of each feature. If *color* and *shape* are the features of an object in the domain, $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$, with color and shape feature instances being members of $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively, e.g., $\mathcal{F}_1 \ni f = blue$ and $\mathcal{F}_2 \ni f = rectangle$. We denote the scene objects with labeled feature$f \in \mathcal{LF}_i$ as $\mathcal{LSF}_i$ and the scene objects with unlabeled features $f \in \mathcal{UF}_i$ as $\mathcal{USF}_i$, such that:

$$|\mathcal{LSF}_i| + |\mathcal{USF}_i| = |\mathcal{LS}| + |\mathcal{US}| = |\mathcal{S}| = m$$

for each $i \in [1, n]$. Note that $\mathcal{LSF}_i \cup \mathcal{USF}_i = \mathcal{S}$ and $\mathcal{LSF}_i \cap \mathcal{USF}_i \equiv \varnothing$ for each $i \in [1, n]$. The ratio of the number of instances of each feature or object the learner will acquire labels for (by posing a specific query) against all the knowledge the learner can acquire about the scene, is denoted by:

$$P(\mathcal{F}_i) = \frac{|\mathcal{LSF}_i|}{m}, \qquad P(\mathcal{S}) = \frac{|\mathcal{LS}|}{m}$$

The information gain (β) is then measured as the product of quantities computed above:

$$\beta = \prod_{i=1}^{n} P(\mathcal{F}_i) \times P(\mathcal{S})$$

which represents the potential *information gain* upon obtaining the answer to candidate query q.

2) Unambiguity: For an object or feature, the learner must also determine how much contextual information should be included in the query to describe the object unambiguously. If there is no unique scene object that satisfies the contextual information embedded in a query, the query becomes ambiguous. The proposed *unambiguity measure* γ captures this reasoning using a modified Chi-square probability distribution with degree of freedom $k = 2$ and $x \in [0, +\infty]$ denoting the number of objects or feature instances in the scene which satisfy the query context:

$$\gamma = f(x) = \begin{cases} x = 0, & 0 \\ x \geq 1, & \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})}(x-1)^{\frac{k}{2}-1}e^{-\frac{x-1}{2}} \end{cases}$$

This distribution that can be simplified for $k = 2$ as:

$$\gamma = \frac{1}{2}e^{-\frac{x-1}{2}}$$

where Γ is the Gamma function such that $\Gamma(1) = 1$.

3) Combined score: The candidate queries are to be ranked in decreasing order of *utility* based on the combination of the measures: information gain, unambiguity, and human confusion. First the utility δ of a query is computed as the product of the information gain and unambiguity measures described above, i.e., $\delta = \beta \times \gamma$. Based on the value of this score, the queries can be ranked relative to each other. The query with a higher δ is preferred for soliciting information from a human. If there are multiple queries with the same δ, a measure of human confusion (α) is used to break the tie. We introduce a simple definition of α based on the level of context information: *Level 1* queries are preferred

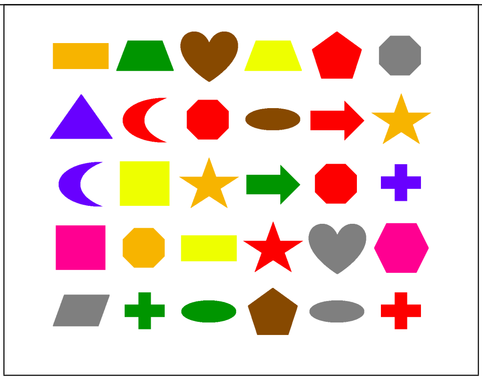

Fig. 2. An example scene with simulated objects.

over *Level 2* queries, and *Level 2* queries will be preferred over *Level 3* queries. The levels of context are described in Section III-A.1. If multiple queries still have the same overall score, one of these queries will be selected randomly.

IV. Experimental Results

This section describes the experimental setup (Section IV-A) and summarizes the results of experimentally evaluating the algorithms described above (Section IV-B).

We report results of evaluating our architecture in a simulated domain, which is simplistic enough to support thorough analysis of the algorithm and the measures, while also capturing the research challenges that are intrinsic to more complex human-robot collaboration domains [13]. The simulated domain abstracts away the uncertainty in the information extracted from visual cues and verbal cues using sensor input processing algorithms. The uncertainty that may exist in human input is also not considered. The experimental results reported below correspond to trials with simulated images of scenes, with objects characterized by color and shape features. The labels of interest therefore include the color labels, shape labels, and object labels[2].

A. Experimental setup

For objects characterized by specific colors and shapes, the feature set $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$, where $\mathcal{F}_1 \ni f = \langle label, RGB \rangle$ is a tuple of color labels and RGB values. Ten different colors are considered in the trials: *Blue, Brown, Grey, Green, Orange, Pink, Red, Yellow, White and Black*. The representations for *White* and *Black* are assumed to be always known; they are the foreground and background colors. Next, $\mathcal{F}_2 \ni f = \langle label, contour \rangle$ is a tuple of shape labels and shape contours; a contour is a set of points on a plane. The 15 shapes in the domain are: *Arrow, Circle, Cross, Heart, Hexagon, Moon, Octagon, Oval, Parallelogram, Pentagon, Rectangle, Square, Star, Trapezoid, Triangle*. A *Scene* is thus a set of colored shapes without any occlusion—Figure 2 shows an example.

The set of relations $\mathcal{R}$ considered in this simulated domain are spatial relationships that are known to the robot and are defined in terms of the known (x, y) coordinates of the centroids of objects, e.g., *above/up/on top* refers to centroids' relative locations in terms of the y axis. The experimental setup allows no more than two spatial relationships to exist between any two objects. The objective of the learner is to learn the labels of the objects and features in the scene by posing as few questions as possible. The number of questions posed is used as the performance measure.

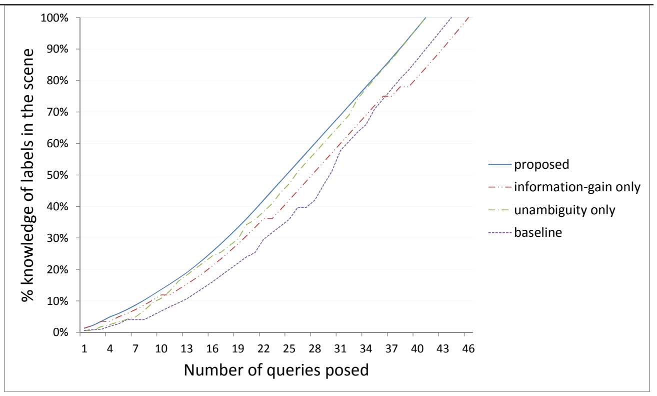

Fig. 3. Knowledge of object and scenes labels expressed as a function of the number of queries posed to obtain this knowledge, for the scene in Figure 2. The proposed query selection algorithm acquires knowledge faster than algorithms that use just the information gain measure or the unambiguity measure, or a random selection strategy.

B. Experimental Results

For the scene in Figure 2, assume that the robot's initial knowledge includes the color labels, shape labels, and object labels of the following four objects: *pink star*, *green arrow*, *blue heart*, and *yellow cross*; not all these objects are in the scene in Figure 2. The following are a subset of the questions generated by the system. Each line starts with the iteration number; the question is either rejected as being ambiguous, or ends with the answer obtained:

- *Iteration 1:* "What is the label of the shape on bottom left of the scene?" **Parallelogram**.
- *Iteration 2:* "What is the label of the object with green color?" **Ambiguous Query**.
- *Iteration 3:* "What is the label of the color of *Parallelogram* shaped object?" **Gray**.

Note that when queries refer to more than one object, e.g., the ambiguous question in the second iteration above, it is not posed. Overall, the system incrementally obtains the necessary information by building on the existing knowledge, and using it to pose the subsequent questions.

Next, Figure 3 compares the proposed query (ranking and) selection algorithm with three other algorithms on the scene illustrated in Figure 2: (1) using only the information gain measure; (2) using only the unambiguity measure; and (3) a baseline approach that randomly selects queries from the candidate set. Figure 3 plots the % knowledge of object and feature labels in the scene as a function of the number of queries posed to acquire this knowledge. Since the proposed query selection algorithm combines *information gain* and *unambiguity measure* to select high utility queries from $\mathcal{Q}$, it provides the best performance with the least number of

[2] In the examples reported here, the object labels are a combination of the color labels and shape labels, but this is not a requirement.

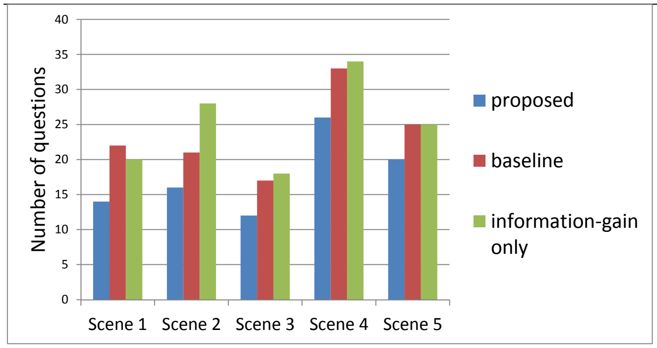

Fig. 4. Evaluation of different scenes in terms of the number of queries required to learn the color, shape and object labels in each scene. Proposed query selection algorithm requires significantly fewer number of queries than the baseline algorithm or the algorithm that only uses the information gain measure to select queries.

queries posed. In contrast, the baseline approach chooses queries randomly from $\mathcal{Q}$, and requires the maximum number of queries to acquire knowledge of object and feature labels in the scene. If an ambiguous query is posed to the annotator, the interaction is considered unsuccessful and leads to no answer. This allows the query selection algorithm that only uses the *unambiguity measure* (Section III-B.2) to obtain complete knowledge of the scene by posing the same (total) number of queries as the proposed query selection algorithm. However, the proposed algorithm allows the robot to maximize the amount of knowledge acquired during each human interaction in the intermediate stages. Since the algorithm that only uses the information gain measure poses ambiguous queries (similar to the random query selection algorithm), it often results in unsuccessful interactions; in fact, using just the information gain measure can be worse than selecting queries randomly. We anticipate the improvement provided by the proposed algorithm is likely to be more pronounced in more complex scenes, especially if the uncertainty in sensor input processing or human feedback is not abstracted away.

Finally, Figure 4 summarizes the results for a set of five scenes. These scenes differ in complexity, i.e., in terms of the number and type of objects in the scene. For each of these scenes, the robot started with the same initial knowledge about a subset of objects in the scene, i.e., labels of these objects and their color and shape features. The proposed algorithm for selecting questions from the set of candidate questions $\mathcal{Q}$ was compared with the algorithm that selected questions randomly from $\mathcal{Q}$ ("baseline"), and with the algorithm that only used the information gain measure to select queries from $\mathcal{Q}$. We observe that for each set of paired experimental trials, our query selection algorithm enables the robot to learn the desired labels of scene objects and features by posing a much smaller number of queries. Over a set of 100 different (randomly generated) scenes with different number and type of objects, the ratio of the average number of questions posed using just the information gain measure with the number of questions posed using our algorithm is 1.19 ± 0.112; the ratio when the random selection algorithm is compared with our algorithm is 1.17 ± 0.106. These results are statistically significant, and are more pronounced as the scenes become more complex, e.g., the ratio is as high as 1.75 in certain scenes when only the information gain measure is used to select questions.

V. Conclusion

To collaborate with humans in complex domains, robots typically need a significant amount of domain knowledge. However, humans may lack the time and expertise to provide accurate domain knowledge or elaborate feedback. The architecture described in this paper generates candidate queries using contextual cues, and combines heuristic measures of information gain, ambiguity, and human confusion, to rank queries based on their relative utility. The top-ranked queries are used to solicit human feedback, which is used to incrementally revise the domain knowledge and pose subsequent queries. Experimental results in a simulated domain indicate that the proposed algorithm and measures significantly reduce the number of queries posed in comparison with a baseline algorithm that selects questions randomly, or uses the individual measures. Future work will consider other types of queries, and model the uncertainty in human feedback and in information extracted from sensors and actuators on physical robots interacting with humans.

References

[1] B. Settles, *Active Learning*, R. J. Brachman, W. W. Cohen, and T. Dietterich, Eds. Morgan & Claypool publishers, 2012.

[2] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2001, pp. 107–118.

[3] C. Zhang and T. Chen, "An active learning framework for content-based information retrieval," Carnegie Mellon Unversity, Pittsburgh, PA, U.S.A, Tech. Rep., 2002.

[4] H. Raghavan, O. Madani, and R. Jones, "Active learning with feedback on both features and instances," *Journal of Machine Learning Research*, vol. 7, pp. 1655–1686, 2006.

[5] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Curran Associates, Inc., 2008, pp. 1289–1296.

[6] K. Salmani and M. Sridharan, "Multi-instance active learning with online labeling for object recognition," in *27th International Conference of the Florida AI Research Society*, 2014.

[7] R. Swaminathan and M. Sridharan, "Towards robust human-robot interaction using multimodal cues," in *Human-Agent-Robot Teamwork Workshop at the International Conference on Human-Robot Interaction*, 2012.

[8] S. Rosenthal, A. K. Dey, and M. Veloso, "How robots' questions affect the accuracy of the human responses," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009, pp. 1137–1142.

[9] M. Gervasio, E. Yeh, and K. Myers, "Learning to ask the right questions to help a learner learn," in *16th International Conference on Intelligent User Interfaces*. ACM, 2011, pp. 135–144.

[10] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, May 2009.

[11] E. A. Billing and T. Hellstrom, "A formalism for learning from demonstration," *Journal of Behavioral Robotics*, 2010.

[12] M. Cakmak and A. Thomaz, "Designing robot learners that ask good questions," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2012, pp. 17–24.

[13] B. Myagmarjav and M. Sridharan, "**(Extended Abstract)** incremental knowledge acquisition with selective active learning," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 4-8, 2015.